

**CST8390 - Business Intelligence and Data Analytics**

**Lab 7 - *Outlier Detection***

**Name: Min Li Id:040930563**

**Due Date:** Week 10 in corresponding lab sessions

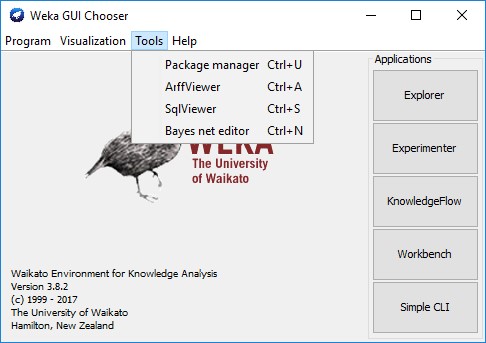
**Introduction**

The goal of this lab is to perform **outlier detection** on Salary File using Local Outlier Factor and Isolation Forest.

**Steps**

# Local Outlier Factor

1. With **Weka** 3.8, outlier detection methods like Local Outlier Factor and Isolation Forest are not included. But they are **available** as packages to be installed using **Package Manager** of **Weka**.



From the package manager, **install** **localOutlierFactor** and **isolationForest**

(find the package in the big list, select it, and hit install).

1. Now, **download** **EmployeesSalaryOriginalOutlier.csv** file from **Brightspace** and **load** it into Weka explorer. If everything worked well, you should be able to see Local Outlier Factor and Isolation Forest listed as classifiers under weka  classifiers  misc on Classify tab.

1. Make sure that **all attributes** are loaded with right data types. If not, **apply filters** to convert them. Save the file as **EmployeesSalaryOriginalOutlier.arff**.

(Expectation: ID, first\_name, last\_name, email, Address - String,

Country, Branch and Currency, Outlier – Nominal

and salary - Numeric).

1. We are going to perform **outlier detection** on this file. There are some attributes that are **not relevant** for outlier detection. Identify and **remove** those attributes. List the names of **removed** attributes below:

|  |
| --- |
| ID, first\_name, last\_name, email, Address |

1. Run addID filter to create an **ID** column.

1. Implementation of outlier detection methods in **Weka** needs a **class attribute**. So, we will use Outlier as the class attribute. In order to detect outliers using Local Outlier Factor, you need to select it from weka  classfiers  misc on Classify tab. You need to select **10-fold** cross validation and Outlier as the class **attribute**.

1. Right click on the result in the result pane and click on “Visualize classifier errors” and save the file as **LOF\_Results.arff**.

1. Now, open another explorer and open **LOF\_Results.arff**. Two more **attributes** are created by LOF, namely prediction margin and predicted outlier. You have a few instances predicted as outliers. Hit Edit to open Viewer. **Sort** Predicted Outlier and see **how many** of actual outliers are predicted as outliers. \_\_\_\_\_\_\_\_5\_\_\_\_\_\_\_.

# Isolation Forest

1. Open **another** explorer and load **EmployeesSalaryOriginalOutlier.arff** from step 3. Remove all irrelevant attributes. Make sure you have the right data types.

1. **Convert** all nominal attributes except Outlier to **binary** using filter.

1. Run addID **filter** to create an ID column.

1. **Run** Isolation Forest by setting “Use training set” as the test option and Outlier as the class **attribute**.

1. **Right click** on the result in the result pane and click on “Visualize classifier errors” and save the file as **ISF\_Results.arff**.

1. Now, open **another explorer** and open **ISF\_Results.arff**. Two more **attributes** are created by LOF, namely prediction margin and predicted outlier. You have a few instances predicted as outliers. Hit Edit to open Viewer. **Sort** Predicted Outlier and see **how many** of actual outliers are predicted as outliers. \_10\_.

# Combine Results

1. Open **EmployeesSalaryOriginalOutlier.csv** and save it as **Results.xlsx**.

1. Open both results file in **Notepad++**. Copy results from LOF\_Results into **another sheet**. Use text to columns to convert data into columns. Add **header** row based on the header info in the **arff** file. Give LOF prefix for the new columns created. **Sort** it based on the ID column. Copy and paste **new columns** into the first sheet of **Results.xlsx**.

1. Next, copy results from **ISF\_Results** from **arff** file into **another sheet** and do the same as in step 13. Give **ISF** prefix for new columns created. Copy and paste new columns into the first sheet of **Results.xlsx**.

1. Now you have both results along with the data in one sheet. **Replace** all Yes with 1 and No with 0 (use find & replace).

1. Create a **new column** named **Ensemble** and apply **formula** that calculates the **sum** of LOF:

predicted Outlier and ISF: predicted Outlier for this column.

1. Select the sheet and **sort** it from Largest to Smallest based on Ensemble column. Your header row of combined sheet should look like:



1. Create a **new column** named **Reason** and record the reason for the instances to be predicted as outlier **based on your judgement**.

**REMEMBER:**

Show your **answers** to the lab professor when you are done.

You should be ready with your explorers for LOF, ISF, LOF results, ISF results and the Results excel file.

***FOR YOUR ANALYSIS:***

* ***Option 1****: Use your own words to explain* ***Local Outlier Factor*** *and situations to use it.*
* ***Option 2****: Use your own words to explain* ***Isolation Forest*** *and situations to use it.*

Options 2: The algorithm isolates each point in the data and splits them into outliers or inliers. This split depends on how long it takes to separate the points. If we try to segregate a point which is obviously a non-outlier, it’ll have many points in its round, so that it will be really difficult to isolate. On the other hand, if the point is an outlier, it’ll be alone and we’ll find it very easily. An advantage of this algorithm is that it works with a huge data set and several dimensions. The dimensions refer to the different features that we have in our data set. The data refers, of course, to each element of the data set.

Local outlier factor. In anomaly detection, the local outlier factor (LOF) is an algorithm proposed by Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jörg Sander in 2000 for finding anomalous data points by measuring the local deviation of a given data point with respect to its neighbours.

Ottawa, Mar 2020.